

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **07-036434**

(43)Date of publication of application : **07.02.1995**

(51)Int.Cl.

G09G 5/24

G06T 11/20

(21)Application number : **05-182976**

(71)Applicant : **HITACHI LTD**

(22)Date of filing : **23.07.1993**

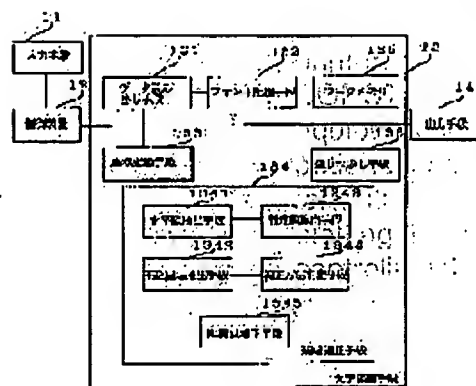
(72)Inventor : **OGAWA NAOMI**

## (54) CHARACTER OUTPUT DEVICE

(57)Abstract:

**PURPOSE:** To correct characters by an optimum method corresponding to the characters by discriminating the correcting method according to characters of the characters and characteristics of a printer.

**CONSTITUTION:** The character output device, equipped with an input means 11 for inputting optional characters and sizes, an output means 14 for outputting them, a character plotting means 13 for plotting the characters, and a control means 12 for controlling them, consists of a data read means 131 which reads character outline coordinate data corresponding to a character code sent from the input means 11 out of a font storage means 132 stored with character outline coordinate data forming an outline font, a coordinate converting means 133 which converts the coordinates of the character outline coordinate data according to specific character size, a painting-out means 136 which paints out the inside of the coordinate-converted character outline coordinate data, a line width correcting means 134 which corrects line width by using the character outline coordinate data whose coordinates are converted by the coordinate converting means 133, and a correcting method selecting means 1344 which selects a correcting method for correcting the line width by the line width correcting means 134.



BEST AVAILABLE COPY

## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]  
 [Patent number]  
 [Date of registration]  
 [Number of appeal against examiner's  
 decision of rejection]  
 [Date of requesting appeal against  
 examiner's decision of rejection]  
 [Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

2005/11/11  
 14:58

2005/11/11  
 14:58

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-36434

(43) 公開日 平成7年(1995)2月7日

(51) Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 9 G 5/24		9471-5G		
G 0 6 T 11/20		9192-5L	G 0 6 F 15/ 72	3 5 5 U

審査請求 未請求 請求項の数8 O L (全 9 頁)

(21) 出願番号 特願平5-182976

(22) 出願日 平成5年(1993)7月23日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 小川 直美

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(74) 代理人 弁理士 武 顕次郎

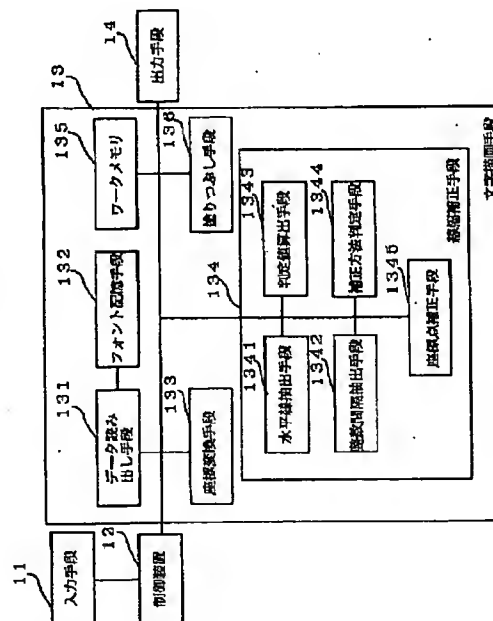
(54) 【発明の名称】 文字出力装置

(57) 【要約】

【目的】 文字の特性及びプリンタの特性により補正方法を区別し、各文字に見合った最適な方法で補正できるようにする。

【構成】 任意の文字及びサイズを入力する入力手段11と、文字を出力する出力手段と14、文字を描画する文字描画手段13と、これらを制御する制御手段12とを備えた文字出力装置において、上記文字描画手段13を、アウトラインフォントを形成する文字輪郭座標データを記憶するフォント記憶手段132から、前記入力手段11より送られてくる文字コードに対応する文字輪郭座標データを読み出すデータ読み出し手段131と、前記文字輪郭座標データを所定の文字サイズに従って座標変換する座標変換手段133と、座標変換された文字輪郭座標データの内部を塗りつぶす塗りつぶし手段136と、前記座標変換手段133において座標変換された文字輪郭座標データを用いて線幅を補正する線幅補正手段134と、前記線幅補正手段134で補正する補正方法を選択する補正方法選択手段1344とから構成した。

図1



1

【特許請求の範囲】

【請求項1】 任意の文字及びサイズを入力する入力手段と、文字を出力する出力手段と、文字を描画する文字描画手段と、これらを制御する制御手段とを備えた文字出力装置において、

上記文字描画手段が、アウトラインフォントを形成する文字輪郭座標データを記憶するフォント記憶手段と、

前記入力手段より送られてくる文字コードに対応する文字輪郭座標データを前記フォント記憶手段から読み出すデータ読み出し手段と、

データ読み出し手段によって読み出された文字輪郭座標データを所定の文字サイズに従って座標変換する座標変換手段と、

座標変換手段によって座標変換された文字輪郭座標データの内部を塗りつぶす塗りつぶし手段と、

あらかじめ設定された複数の補正方法の中から補正する方法を選択する補正方法判定手段を含み、当該補正方法判定手段によって選択された補正方法により前記座標変換された文字輪郭座標データを用いて線幅を補正する線幅補正手段と、を備えていることを特徴とする文字出力装置。

【請求項2】 前記補正方法判定手段が、任意の条件のもとで補正方法を選択するように設定され、

前記線幅補正手段が、前記座標変換手段により変換された文字輪郭座標データから水平線輪郭座標のy座標および垂直線輪郭座標のx座標を抽出する線抽出手段と、

抽出された前記線輪郭座標のy座標および垂直線輪郭座標のx座標の各々に対し、隣接した座標値の間隔を求め、該間隔をそれぞれ整数化する整数間隔抽出手段と、

前記補正方法判定手段における判定の基準となる値を算出する判定値算出手段と、

前記補正方法判定手段により補正された間隔の和を求めることにより前記輪郭座標を再計算し、補正を行う座標点補正手段と、を備えていることを特徴とする請求項1記載の文字出力装置。

【請求項3】 前記補正方法判定手段によって選択される補正方法が、文字の特性を基準にして選択される2つの補正方法であることを特徴とする請求項1または2記載の文字出力装置。

【請求項4】 前記2つの補正方法が、線を太めに補正する方法と細めに補正する方法であることを特徴とする請求項3記載の文字出力装置。

【請求項5】 前記任意の条件が、文字の複雑さ及びサイズを基準に設定されていることを特徴とする請求項2記載の文字出力装置。

【請求項6】 前記任意の条件が、文字の境界ボックスの中の白領域の面積を基準に設定されていることを特徴とする請求項2記載の文字出力装置。

【請求項7】 前記任意の条件が、座標データを整数化したものと線幅を整数化したものとの誤差を基準に設定

2

されていることを特徴とする請求項2記載の文字出力装置。

【請求項8】 前記任意の条件が、線幅を整数化したときに生じる丸め誤差を基準に設定されていることを特徴とする請求項2記載の文字出力装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、レーザ・ビーム・プリンタ、ディスプレイ装置などの表示装置に係り、特にマルチサイズ、マルチフォントの高品質文字を含む文書を作成する文字出力装置に関する。

【0002】

【従来の技術】レーザ・ビーム・プリンタ、ディスプレイ等の表示装置における文字データの記憶形式としてアウトラインフォントがある。アウトラインフォントは、文字の形状を文字の輪郭の座標列として記憶するものである。このアウトラインフォントを印字するには、まず所望の印字サイズにするために各輪郭座標値を座標変換し、その後、変換した輪郭座標の内部を塗りつぶすという処理を行う。アウトラインフォントは、拡大縮小回転等が自由自在にできるという特長があるが、小さな文字を印字する場合には、量子化誤差により文字の垂直線・水平線の線幅が不揃いとなり、文字の表示品質が低下するという問題があった。

【0003】この線幅を補正するための従来方法として、特開平4-188190公報記載のものが知られている。

【0004】この方法は、座標変換後の輪郭座標から水平・垂直線の輪郭座標を抽出し、輪郭座標間の間隔を整数化し、得られた整数間隔をもとに輪郭座標を再計算する方法であり、線幅補正のための付加的な情報を持たなくても任意の文字サイズに対して線幅補正を可能にしていた。

【0005】

【発明が解決しようとする課題】しかし、上記従来方法は量子化誤差による線幅のバラツキを補正することに重点を置き、各文字に対する細かな補正について配慮されていなかった。すなわち、画数の多い複雑な文字、単純な文字、さらに文字サイズの大小などの文字の諸要素の相違にかかわらず同じ補正方法をとっていた。そのため、画数の多い複雑な文字や小さい文字に対してはつぶれがなくなり、本来のアウトラインフォントの美しさを十分に表現できるだけの成果を上げていたが、画数の少ない単純な文字は線が細めになり、全体的に弱々しい印象があった。

【0006】これを図5に基づいて具体的に説明する。この例では、「日」という文字をアウトラインフォントによって処理する場合を示している。すなわち「日」という文字の各輪郭の水平方向の輪郭値を抽出すると、①に示すように下から「0.00」、「6.36」、「1

10

20

30

40

50

3

0、60」、「25、44」、「29、68」、「44、52」、「48、76」となる。これを補正処理を行わないで50×50ドットの文字として出力するために整数化処理すると⑦に示すように「0」、「6」、「11」、「25」、「30」、「45」、「49」となる。すると、線幅はそれぞれの差をとって、⑧に示すように下から「5」、「5」、「4」となる。すなわち一番上の線が20%だけ他の線よりも細くなる。このように細くなると、線のバランスが崩れ、文字の美しさが阻害されるので、線幅を揃えるように補正するようになった。これが上記従来例である。

【0007】そこで、この従来例では、①で検出した水平線の輪郭値の差をとって②のようにして各線幅を算出し、さらにその幅を四捨五入して整数化処理すると③で示すように線間隔「6」、「4」、「15」、「4」、「15」、「4」を得る。しかし、このようにして得られた線間隔は、合計すると「48」となるが、②における抽出された線幅算出値を合計すると「48、76」で、これは整数化処理すると四捨五入して「49」となる。したがって、線幅を算出したものとドットに変換したものとでは「1」ドットのずれが出てくる。そこで、この補正では、④のように線幅が細くならないように線と線との間を補正し、④に示すように下側の空白部分を「15」ドットから「16」ドットにした。そして、⑤において座標点を補正し、⑥に示すように「4」ドットずつの線幅の文字として出力し、線幅を揃えた綺麗な文字としている。

【0008】このように補正した実例を図6に示す。図6は400DPIのレーザ・ビーム・プリンタによって出力された1字のドット数が50×50ドットの文字の拡大図で、同図(a)は上述のようにして補正した「信」の文字である。言い換えれば⑥のようにした「信」の文字であり、同図6(b)は補正しない場合の「信」の文字で、図5では⑥に相当する。このように補正すると、線が細くなって弱々しい印象を受けることは否めない。

【0009】本発明は、このような背景に鑑みてなされたもので、その目的は、文字の特性及びプリンタの特性により補正方法を区別し、各文字に見合った最適な方法で補正できるような文字出力装置を提供することにある。

【0010】

【課題を解決するための手段】上記目的を達成するため、本発明は、文字の複雑さやプリンタの特性等により補正方法を制御するように構成されている。例えば、プリンタの解像度に基づいた文字のドットサイズ、白領域の幅等のように簡単な条件のもとで補正方法を振り分ける。その結果、各文字に見合った方法で補正を行うことができるため、複雑な文字はつぶれないように細めに補正を行い、単純な文字、すなわちつぶれる可能性のない

4

文字に対しては、弱々しくならないよう太めに補正するといったように、本来の文字の形状を崩さず、各文字に適した補正が可能となる。

【0011】具体的には、任意の文字及びサイズを入力する入力手段と、文字を出力する出力手段と、文字を描画する文字描画手段と、これらを制御する制御手段とを備えた文字出力装置において、上記文字描画手段を、アウトラインフォントを形成する文字輪郭座標データを記憶するフォント記憶手段から、前記入力手段より送られてくる文字コードに対応する文字輪郭座標データを読み出すデータ読み出し手段と、前記文字輪郭座標データを所定の文字サイズに従って座標変換する座標変換手段と、座標変換された文字輪郭座標データの内部を塗りつぶす塗りつぶし手段と、あらかじめ設定された複数の補正方法の中から補正する方法を選択する補正方法判定手段を含み、当該補正方法判定手段によって選択された補正方法により前記座標変換された文字輪郭座標データを用いて線幅を補正する線幅補正手段とから構成した。

【0012】

【作用】入力手段により文字及びサイズが入力されると、制御手段は、フォントデータ記憶手段からアウトラインデータを読み出し、座標変換を行う。次に線幅補正手段では、座標変換された文字輪郭座標データから水平線輪郭座標のy座標、垂直線輪郭座標のx座標を抽出し、抽出された各々の座標に対し、隣接した座標値の間隔を求め、該間隔を整数化し、例えば文字の線間の白領域の幅のような補正方法判定手段で用いる判定値を算出する。そして判定値を用いて該文字に適した補正方法を選択し、補正を行う。補正が終了した後、前記補正に従い、前記輪郭座標を再計算して座標点を補正し、出力手段から文字を出力させる。

【0013】また、例えば線幅太め補正及び線幅細め補正の2種類の補正方法をあらかじめ設定しておき、文字及びプリンタの特性に従って最適と思われる補正方法を割り付ければよいので、複雑な文字であってもサイズが大きければ、太めの補正処理を、小さければ細め（従来）の補正処理を、というように、文字本来の形状を崩さず、サイズを考慮した上で、該文字に見合った補正が可能となる。ここで、本来同じ線幅をもつ複数の線の複数组存在する場合、そのうち1組でも太めに補正できる線が存在すれば太め補正を行うことになる。

【0014】

【実施例】以下、図面を参照し、本発明の実施例について説明する。

【0015】図1は、本発明の実施例に係る文字出力装置のシステム構成を示すブロック図である。同図において、このシステムは、文字を入力する入力手段11と、一連の処理を制御する制御装置12と、文字を描画する文字描画手段13と、文字を出力する出力手段14とから基本的に構成されている。

【0016】文字描画手段13は、フォントデータを記憶しておくフォント記憶手段132と、フォント記憶手段132からフォントデータを読み出すデータ読み出し手段131と、読み出したフォントデータから所定の文字サイズに変換する座標変換手段133と、変換された文字輪郭座標データから線幅を補正する線幅補正手段134と、読み出された文字輪郭座標データ等の計算処理を行うワークメモリ135と、文字イメージを生成するために文字輪郭内部を塗りつぶす塗りつぶし手段136とからなっている。さらに、線幅補正手段134は、文字輪郭座標データから水平線及び垂直線を抽出し、各線幅を算出する水平線抽出手段1341と、算出された各線幅を整数化して整数間隔を求める整数間隔抽出手段1342と、各文字に適した補正方法を割り当てる際の判定に用いる値を算出する判定値算出手段1343と、判定値算出手段1343により算出した値を用いて補正方法を判定する補正方法判定手段1344と、補正方法判定手段1344により選択された補正方法に基づき、座標点の補正を行う座標点補正手段1345とからなっている。

【0017】このように構成されたシステムにおいては、入力手段11は文字描画手段13に印字命令を転送すると、文字描画手段13は前記印字命令を受けて文字イメージを生成し、出力手段14に転送する。これら一連の処理を制御するのは制御装置12である。出力手段14はプリンタ、ディスプレイ等であり、前記文字イメージを印刷もしくは表示する。文字描画手段13の内部では、データ読み出し手段が入力手段11から送られてきた書体情報と文字コードを解読し、指定された文字輪郭座標データをフォント記憶手段132からワークメモリ135に読み出す。座標変換手段133は入力手段11より文字サイズ情報を受け取り、ワークメモリ135に読み出された文字輪郭座標データを所定のサイズになるように変換する。このとき、文字輪郭座標データがベジェ曲線等のような曲線表現を含む場合には、座標変換手段133において直線近似処理を行う。変換を受けた文字輪郭座標データは線幅補正手段134において補正される。塗りつぶし手段136は補正された文字輪郭座標データをもとに、輪郭内部を塗りつぶすことによって文字イメージを生成し、出力手段14に転送する。

【0018】線幅補正手段134の内部では、まず水平線抽出手段1341が座標変換手段133により変換を受けた文字輪郭座標データから水平線・垂直線を抽出し、その線幅を算出する。整数間隔抽出手段1342では、算出した線幅から各線幅を整数化し、前記判定値算出手段1343では、前記水平線抽出手段1341及び前記整数間隔抽出手段1342で算出した各々の線幅等の情報以外に補正方法を判定する際に必要となる値があれば算出し、補正方法判定手段1344では該文字に適した補正方法を選択する。座標点補正手段1345で

は、前記補正方法判定手段1344で選択した補正方法に基づき、座標点の補正を行う。

【0019】引き続き図2ないし図4を参照して線幅補正処理の処理手順について説明する。

【0020】図2は前記線幅補正手段134における処理手順を示すフローチャートであり、図3は補正の手順を示す説明図である。この補正では、まず、ステップST21において、読み出された文字輪郭座標データより水平線輪郭座標( $y_0 \sim y_7$ )を抽出する。次いで、ステップST22で、ステップST21で抽出した水平線間の領域が文字を構成する線分(黒領域)であるか文字の線間の部分(白領域)であるかを判定し、これを白黒領域判定テーブル31に記憶する。この判定の方法の一例として特開平4-188190公報記載のものが知られているが、判定方法は上記公報記載以外の方法でも構わないことはもちろんである。このようにして白黒領域を判定すると、ステップST23で、ステップST22で求めたすべて白黒領域の間隔( $dy_0 \sim dy_6$ )を抽出し、これを水平線間隔テーブル32に記憶する。さらにこれらの和 $y_t 34$ を求め、さらに $y_t 34$ を整数化した $i y_t 36$ を求めておく。

【0021】ステップST23で線間隔を抽出した後、ステップST24で、ステップST23で算出した線間隔を整数化し、各々の整数間隔( $idy_0 \sim idy_6$ )を抽出し、水平線整数間隔テーブル33に記憶する。さらにこれらの和 $i y_s 35$ を求めておく。ステップST25では、補正方法を区別する判定値を算出する。ここで、もし判定値がステップST21~24で既に算出されていれば何もしないことになる。次にステップST26では、ステップST25及びステップST21~24で求めた判定値をもとに補正方法を選択する。もし判定条件を満足する場合にはステップST27の線幅太め補正処理を行い、満足しない場合にはステップST28の線幅細め補正処理、すなわち従来の補正処理を行う。次にステップST29で、判定された補正処理に基づき、ステップST23及びステップST24で求めた $y_t 34$ 、 $i y_s 35$ を用いて、全体長の調整も考慮した座標点の補正を行う。

【0022】上記ステップST25で算出した判定値は、上記ステップST26で補正方法を判定する際に使用される値であるが、ステップST26で補正方法を判定する例を図3及び図4を用いて説明する。

【0023】図4において、ステップST41で、まず黒領域の線間隔( $dyn$ )と整数間隔( $idy_n$ )を比較して、線間隔が整数間隔以上であればステップST42の処理に進み、小さければステップST28へ進む。ステップST42では、ステップST41で比較対象とした黒領域の下に隣接する白領域の線間隔( $dyn-1$ )と整数間隔( $idy_n-1$ )を比較する。ここで整数間隔が線間隔以上であれば、ステップST43の処理

に進み、小さければステップST28へ進む。ステップST43では、ステップST42で対象とした白領域の線間隔( $d_{yn}-1$ )が2ドット以上のとき、ステップST27へ進んで線幅太め補正処理を行い、小さければステップST28へ進んで線幅細め補正処理を行う。

【0024】本実施例は、座標変換後の座標データより算出した水平線間隔テーブル32中の黒領域の線間隔で、水平線整数間隔テーブル33中の整数化した整数間隔以上であるものが存在し、さらに上記条件を満足する黒領域に隣接する水平線間隔テーブル32中の白領域の線間隔が、対応する水平線整数間隔テーブル33中の整数間隔以下でかつ2ドット以上の幅があるとき、という条件で補正方法を区別するものである。すなわち、この実施例は、文字の複雑さやサイズ、及びプリンタの解像度等を考慮して補正方法を決定する例である。

【0025】さらに図3を用いて別の実施例について説明する。

【0026】この実施例は、座標変換後の座標データを直接整数化し、各白黒領域の線幅を補正方法の選択に用いる。その際に必要となる前記値は判定値算出手段1343であらかじめ求めておく。ここでは座標変換後の座標データを直接整数化して整数座標テーブル37に記憶し、この整数値より線幅を算出して水平線整数間隔座標テーブル38に記憶する。そして水平線整数間隔テーブル33に記憶してある値( $i_{dyn}$ )と水平線整数間隔座標テーブル38に記憶してある値( $i_{dn}$ )を比較し、本来線幅が等しい複数の黒領域で $i_{dn} > i_{dyn}$ を満たす値が存在し、かつこの条件39を満足する値( $i_{dm}$ )の両端の白領域( $i_{dm}-1$ ,  $i_{dm}+1$ )のいずれかが2ドット以上であるという条件40を満足しているとき、太め補正処理41を行う。本実施例は、補正しないで描画するときの座標値と、従来方法による補正を行った後の座標値を比較して、太めに補正できる線が存在すれば太め処理を行うというものである。ここで、整数化される前に同じ間隔であった輪郭座標間の間隔は、整数化後も同じ間隔となるのはもちろんである。

【0027】次いで、さらに他の実施例について図3を用いて説明する。

【0028】本実施例は前記ステップST26中の最初の条件を

(黒領域の各整数線幅の和) + (白領域×2) ≤ 全体長  
とするものである。

【0029】これは、全体長 $i_{ys}35$ からすべての水平線整数間隔テーブル33中の黒領域の整数線幅の和を引いた残りが2ドット以上分の白領域が確保できるという条件である。この条件を満たすことができれば、仮にすべての黒領域を1ドット分増加させても文字のつぶれを引き起こすことはないため、太め処理が可能となる。また、上記条件は太め処理が可能であるかどうかを判定

するのみに留まっている。太め補正が必要である線を実際に抽出するには、上記実施例のような条件をさらに付加することが必要となる。

【0030】次に太め処理における実施例を示す。これは前記判定値算出手段において、線を太めに補正できるという条件を満足した本来同じ線幅の1組の線に対して、1ドット分線幅を増加する。上記以外の処理については、例えば特開平4-188190公報のような従来方法に従って行うことになる。また座標点補正処理は、上記以外の方法であってもよいことはもちろんである。

【0031】これらの実施例によれば、複雑な文字はつぶれをなくすために細めに補正し、単純な文字は弱々しくなるのを防ぐために太めに補正する等、各文字ごとに適した補正方法を選択することが出来る。これら実施例で示したように、補正方法の選択基準は上記の方法でなくともよいことはもちろんである。

【0032】また上記では、水平線輪郭座標のy座標に対する処理のみを説明したが、垂直線輪郭座標のx座標に関しても同様の処理を行う。

【0033】

【発明の効果】これまでの説明で明らかなように、本発明によれば、線幅補正のための付加的な情報を持たないアウトラインフォントに関して、文字の特性及びプリンタの特性により補正方法を区別し、任意の文字及びサイズに適した線幅補正が可能となるので、各文字に見合った最適な方法で補正することができる。

【図面の簡単な説明】

【図1】本発明の実施例に係るシステムを示すブロック図である。

【図2】実施例に係る文字出力処理の処理手順を示すフローチャートである。

【図3】実施例に係る線幅補正処理を示す説明図である。

【図4】他の実施例を示すフローチャートである。

【図5】従来例に係る補正方法を説明するための説明図である。

【図6】未補正の文字と従来例に係る方法で補正した文字を示す図である。

【符号の説明】

- 11 入力手段
- 12 制御装置
- 13 文字描画手段
- 131 データ読み出し手段
- 132 フォント記憶手段
- 133 座標変換手段
- 134 線幅補正手段
- 135 ワークメモリ
- 136 塗りつぶし手段
- 1341 水平線抽出手段
- 1342 整数間隔抽出手段

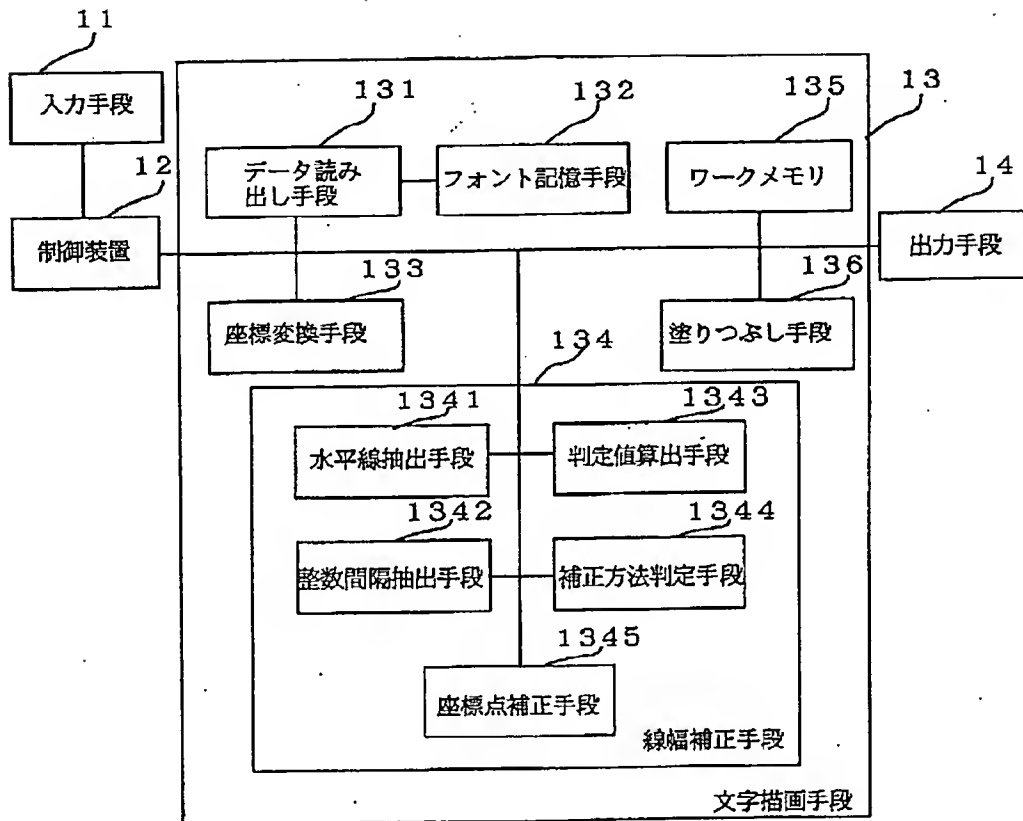
1343 判定値算出手段  
 1344 補正方法判定手段  
 1345 座標点補正手段  
 14 出力手段  
 31 白黒領域判定テーブル

\* 32 水平線間隔テーブル  
 33 水平線整数間隔テーブル  
 37 整数座標テーブル  
 38 水平整数間隔座標テーブル

\*

【図1】

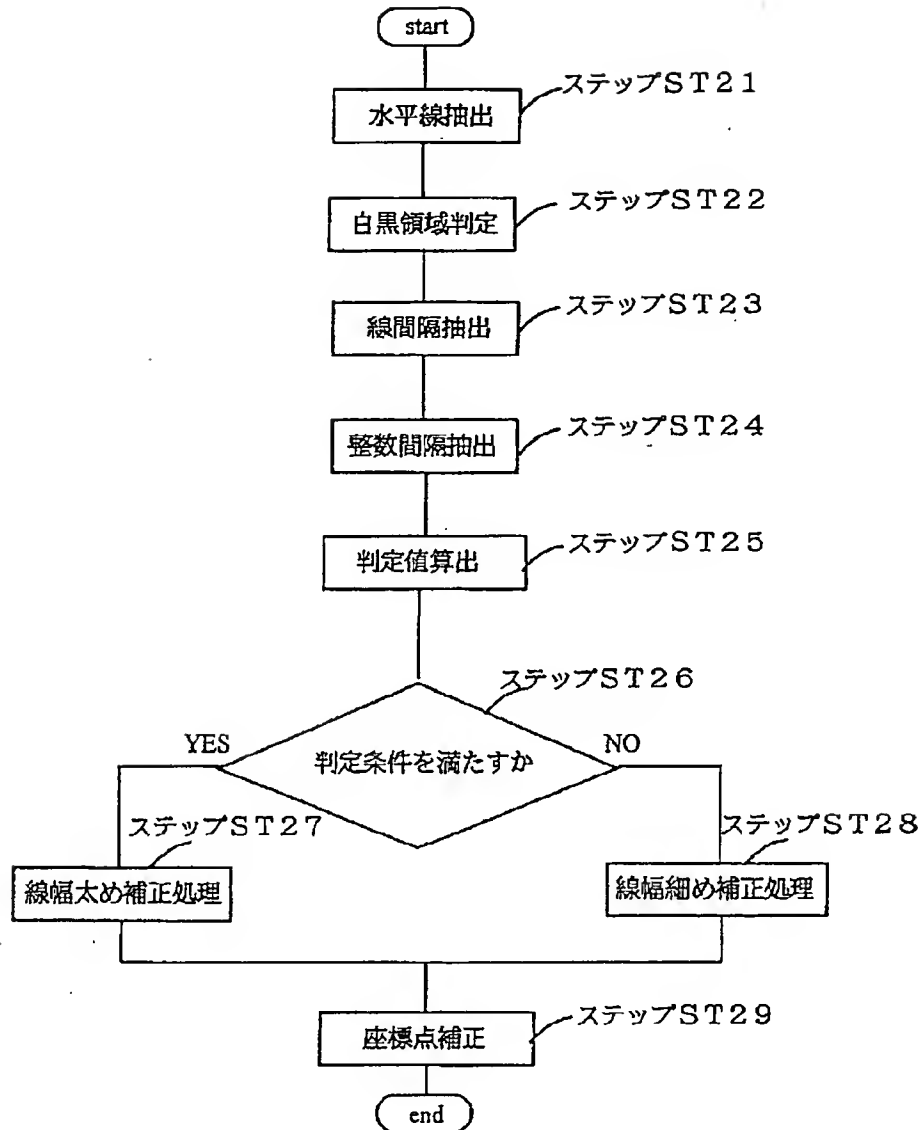
図1





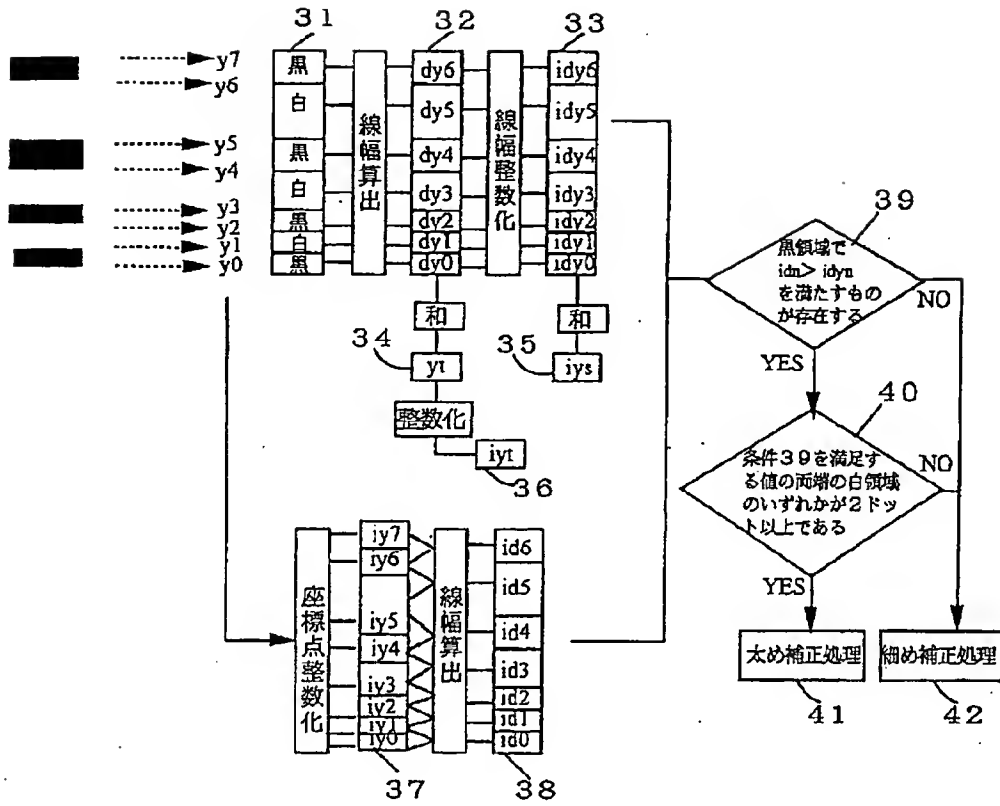
【図2】

図2



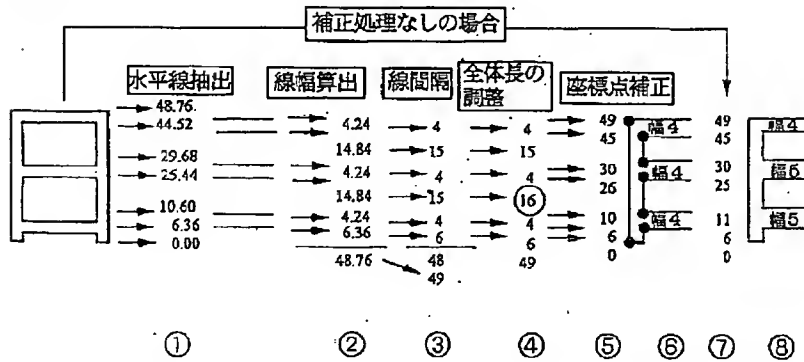
【図3】

図3

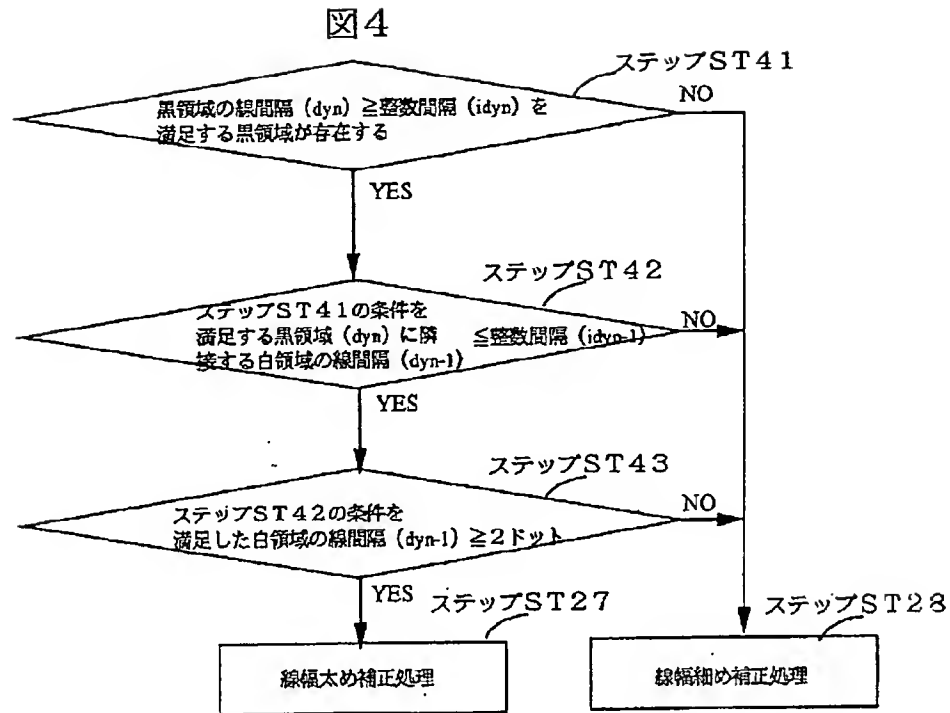


【図5】

【図5】

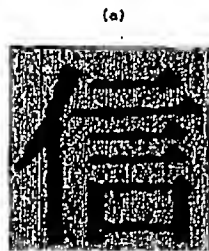


【図4】



【図6】

【図6】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

\* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention relates to the character output device which draws up the document which is applied to displays, such as a laser beam printer and a display device, especially contains the quality character of multi-size and a multifont.

[0002]

[Description of the Prior Art]There is an outline font as a memory form of the alphabetic data in displays, such as a laser beam printer and a display. An outline font memorizes the shape of a character as a coordinate string of the outline of a character. In order to print this outline font, and to use desired print size first, each contour coordinate value is transformed.

Then, processing in which the inside of the changed contour coordinate is smeared away is performed.

Although the outline font had the feature that scaling rotation etc. could be performed freely, when a small character was printed, the line width of the perpendicular line and the horizon of a character became irregular according to the quantization error, and there was a problem that the display quality of a character deteriorated.

[0003]As the conventional method for amending this line width, the thing given in a JP,4-188190,A gazette is known.

[0004]This method is the method of re-calculating a contour coordinate based on the integer interval acquired by extracting the contour coordinate of level and a perpendicular line from the contour coordinate after coordinate conversion, and integer-izing the interval between contour coordinates.

Even if it did not have the additional information for line width amendment, line width amendment was enabled to arbitrary character sizes.

[0005]

[Problem(s) to be Solved by the Invention]However, the above-mentioned conventional method was not considered about the fine amendment to each character with emphasis on amending the variation in the line width by a quantization error. That is, the complicated character with many stroke counts, the simple character, and the still more nearly same correcting method irrespective of a difference of the various elements of characters, such as size of character size, were taken. Therefore, although crushing was lost to the complicated character with many stroke counts, or the small character and only the

success which can fully express the beauty of an original outline font was achieved, the line became thinness and the simple character with few stroke counts had an impression weak-looking on the whole.

[0006]This is concretely explained based on drawing 5. This example shows the case where the character "day" is processed by an outline font. That is, if the horizontal outline value of each outline of the character "day" is extracted, as shown in \*\*, it will be set to "0.00", "6.36", "10.60", "25.44", "29.68", "44.52", and "48.76" from the bottom. Since this is outputted as a character of 50x50 dots without performing a compensation process, if it integer--ization-processes, it will be set to "0", "6", "11", "25", "30", "45", and "49" as shown in \*\*. Then, line width takes each difference, and as shown in \*\*, it is set to "5", "5", and "4" from the bottom. That is, the top line becomes thinner than other lines only 20%. Thus, since the balance of the line collapsed and the beauty of the character was checked when it became thin, it came to amend so that line width may be arranged. This is the above-mentioned conventional example.

[0007]Then, each line width is computed by taking the difference of the outline value of the horizon detected by \*\* in this conventional example, and carrying out like \*\*, and if that width is rounded off and integer--ization-processed further, as \*\* shows, line spacing "6", "4", "15", "4", "15", and "4" will be obtained. However, if the line spacing produced by doing in this way is totaled, it will be set to "48", but when the extracted line width computed value in \*\* is totaled, it is "48.76", and if this is integer--ization-processed, it will be rounded off and will be set to "49." Therefore, a gap of "1" dot comes out by what computed line width, and the thing changed into the dot. So, in this amendment, between a line and lines was amended so that line width might not become thin like \*\*, and as shown in \*\*, the lower blank part was made into "16" dots from "15" dots. And a coordinate point is amended in \*\*, as shown in \*\*, it outputs as a character of the line width of every "4" dots, and it is considered as the beautiful character which arranged line width.

[0008]The example amended in this way is shown in drawing 6. Drawing 6 is an enlarged drawing of the character whose dot number of 1 character outputted by the laser beam printer of 400DPI is 50x50 dots, and the figure (a) is a character of "\*\*\*" amended as mentioned above. In other words, it is a character of "\*\*\*" carried out like \*\*, the drawing 6 (b) is a character of "\*\*\*" when not amending, and it is equivalent to \*\* in drawing 5. If it amends in this way, a line becoming thin and receiving a weak-looking impression cannot be denied.

[0009]This invention was made in view of such a background, and the purpose distinguishes a correcting method with the characteristic of a character, and the characteristic of a printer, and there is in providing the character output device which can be amended by the optimal method corresponding to each character.

[0010]

[Means for Solving the Problem]In order to attain the above-mentioned purpose, this invention is constituted so that a correcting method may be controlled by complexity of a character, the characteristic of a printer, etc. For example, a correcting method is distributed under easy conditions like dot size of a character based on resolution of a printer, and width of a white region. As a result, since it can amend by a method corresponding to each character, a complicated character amends thinness so that it may not be crushed, and a simple character, i.e., a character which must have been crushed, is

received, Amendment suitable for each character is attained without breaking down shape of an original character, as it said that it amended more thickly so that it may not become weak-looking.

[0011] In a character output device specifically provided with an input means which inputs arbitrary characters and sizes, an output means which outputs a character, a character drawing means to draw a character, and a control means which controls these, A data read means which reads character-outline coordinate data corresponding to a character code sent from said input means from a font memory means which memorizes character-outline coordinate data which forms an outline font for the above-mentioned character drawing means, A coordinate transformation means which transforms said character-outline coordinate data according to predetermined character size, A continuous tone means to smear away an inside of character-outline coordinate data by which coordinate conversion was carried out, It constituted from a line width compensation means which amends line width using said character-outline coordinate data by which coordinate conversion was carried out with a correcting method chosen by the correcting method judging means concerned including a correcting method judging means which chooses a method of amending out of two or more correcting methods set up beforehand.

[0012]

[Function] If a character and size are inputted by an input means, a control means will read outline data from a font data memory measure, and will perform coordinate conversion. Next, the y-coordinate of the character-outline coordinate data by which coordinate conversion was carried out in the line width compensation means to a horizon contour coordinate, The decision value which extracts the x-coordinate of a perpendicular line contour coordinate, searches for the interval of the adjoining coordinate value from each extracted coordinates, and integer-izes this interval, for example, is used by a correcting method judging means like the width of the white region between the lines of a character is computed. And it amends by choosing the correcting method which was suitable for this character using the decision value. After amendment is completed, according to said amendment, said contour coordinate is re-calculated, a coordinate point is amended, and a character is made to output from an output means.

[0013] Since what is necessary is just to assign the correcting method which sets up beforehand two kinds of correcting methods, line width thicker amendment and line width thinner amendment, for example, and is considered to be the optimal according to the characteristic of a character and a printer, the compensation process which will narrow a compensation process thicker if size is large even if it is a complicated character if small (former) -- after not breaking down original shape of the character but taking size into consideration so that it may say, amendment corresponding to this character is attained. Here, when two or more sets of two or more lines with the line width same originally exist, thicker amendment will be performed if the line which can be amended more thickly at least 1 set of them exists.

[0014]

[Example] Hereafter, the example of this invention is described with reference to drawings.

[0015] Drawing 1 is a block diagram showing the system configuration of the character output device concerning the example of this invention. In the figure, this system comprises fundamentally the input means 11 which inputs a character, the control device

12 which controls a series of processings, the character drawing means 13 to draw a character, and the output means 14 which outputs a character.

[0016]The font memory means 132 the character drawing means 13 remembers font data to be, The data read means 131 which reads font data from the font memory means 132, The coordinate transformation means 133 changed into predetermined character size from the read font data, It consists of the line width compensation means 134 which amends line width from the changed character-outline coordinate data, the work memory 135 which performs computations, such as read character-outline coordinate data, and a continuous tone means 136 to smear away the inside of a character outline in order to generate a character image. A level line-extraction means 1341 for the line width compensation means 134 to extract the horizon and a perpendicular line from character-outline coordinate data, and to compute each line width, An integer interval extraction means 1342 to integer-ize each computed line width and to search for an integer interval, The decision value calculating means 1343 which computes the value used for the judgment at the time of assigning a correcting method suitable for each character, It consists of the correcting method judging means 1344 which judges a correcting method using the value computed by the decision value calculating means 1343, and the coordinate point compensation means 1345 which amends a coordinate point based on a correcting method with the selected correcting method judging means 1344.

[0017]In the system constituted in this way, the character drawing means 13 will generate a character image in response to said print command, and the input means 11 will transmit it to the output means 14, if a print command is transmitted to the character drawing means 13. The control device 12 controls processing of these series. The output means 14 are a printer, a display, etc., and print or display said character image. Inside the character drawing means 13, a data read means decodes the typeface information and the character code which have been sent from the input means 11, and reads the specified character-outline coordinate data from the font memory means 132 to the work memory 135. The coordinate transformation means 133 receives character size information from the input means 11, and it changes the character-outline coordinate data read to the work memory 135 so that it may become predetermined size. When character-outline coordinate data includes curvilinear expression of a Bezier curve etc. at this time, straight-line approximation processing is performed in the coordinate transformation means 133. The character-outline coordinate data which received conversion is amended in the line width compensation means 134. Based on the amended character-outline coordinate data, by smearing away the inside of an outline, the continuous tone means 136 generates a character image, and transmits it to the output means 14.

[0018]Inside the line width compensation means 134, first, the level line-extraction means 1341 extracts the horizon and a perpendicular line from the character-outline coordinate data which received conversion by the coordinate transformation means 133, and computes the line width. From the computed line width, in the integer interval extraction means 1342, integer-ize each line width, and in said decision value calculating means 1343. It will compute, if there is a value which is needed when judging a correcting method in addition to information, including each line width which were computed by said level line-extraction means 1341 and said integer interval extraction means 1342, and in the correcting method judging means 1344, a correcting method suitable for this character is chosen. In the coordinate point compensation means 1345, a



coordinate point is amended based on the correcting method selected by said correcting method judging means 1344.

[0019]With reference to drawing 2 thru/or drawing 4, the procedure of a line width compensation process is explained succeedingly.

[0020]Drawing 2 is a flow chart which shows the procedure in said line width compensation means 134, and drawing 3 is an explanatory view showing the procedure of amendment. In this amendment, a horizon contour coordinate ( $y_0$ - $y_7$ ) is first extracted from the read character-outline coordinate data in step ST21. Subsequently, it judges whether it is a line segment (black area) in which the field between the horizon extracted by step ST21 constitutes a character from step ST22, or it is a portion (white region) between the lines of a character, and this is memorized on monochrome area judgment table 31. As for the judgment method, although the thing given in a JP,4-188190,A gazette is known as an example of the method of this judgment, it is needless to say that methods other than the above-mentioned gazette statement may be used. thus, when monochrome field was judged, it asked by step ST23 step ST22 -- the interval ( $dy_0$ - $dy_6$ ) of monochrome field is extracted altogether, and this is memorized on the level line spacing table 32. Furthermore these peace  $yt_{34}$  are calculated and  $iyt_{36}$  which integer-ized  $yt_{34}$  further is calculated.

[0021]After extracting line spacing by step ST23, by step ST24, line spacing computed by step ST23 is integer-ized, each integer interval ( $idy_0$ - $idy_6$ ) is extracted, and it memorizes on the horizon integer interval table 33. Furthermore, these peace  $iys_{35}$  are calculated. The decision value which distinguishes a correcting method is computed in step ST25. Here, nothing will not be done if the decision value is already computed by step STs 21-24. Next, in step ST26, a correcting method is chosen based on the decision value calculated by step ST25 and step STs 21-24. In satisfying criteria, it performs the line width thicker compensation process of step ST27, and in not being satisfied, it performs the line width thinner compensation process of step ST28, i.e., the conventional compensation process. Next, the coordinate point which also took adjustment of whole length into consideration by step ST29 using  $yt_{34}$  and  $iys_{35}$  which were calculated by step ST23 and step ST24 based on the judged compensation process is amended.

[0022]Although the decision value computed by the above-mentioned step ST25 is a value used when judging a correcting method by the above-mentioned step ST26, it explains the example which judges a correcting method using drawing 3 and drawing 4 by step ST26.

[0023]In drawing 4, the line spacing ( $d_{yn}$ ) and the integer interval ( $id_{yn}$ ) of a black area are compared first, by step ST41, if line spacing is beyond an integer interval, it will progress to processing of step ST42, and if small, it will progress to step ST28. Step ST42 compares the line spacing ( $d_{yn-1}$ ) and the integer interval ( $id_{yn-1}$ ) of a white region which adjoin under the black area made into the comparison object by step ST41. If an integer interval is more than line spacing here, it will progress to processing of step ST43, and if small, it will progress to step ST28. In step ST43, when the line spacing ( $d_{yn-1}$ ) of the target white region is 2 or more dots in step ST42, it progresses to step ST27 and a line width thicker compensation process is performed, if small, it will progress to step ST28 and a line width thinner compensation process will be performed.

[0024]This example is the line spacing of the black area in the level line spacing table 32 computed from the coordinate data after coordinate conversion, What is beyond the

integer-ized integer interval in the horizon integer interval table 33 exists, The line spacing of the white region in the level line spacing table 32 contiguous to the black area with which it is furthermore satisfied of the above-mentioned conditions is below the integer interval in the corresponding horizon integer interval table 33, and a correcting method is distinguished on the conditions of the time of there being a width of 2 or more dots. That is, this example is an example which determines a correcting method in consideration of the complexity of a character, size, the resolution of a printer, etc.

[0025]Furthermore, another example is described using drawing 3.

[0026]This example forms the coordinate data after coordinate conversion into a direct integer, and uses the line width of each monochrome field for selection of a correcting method. Said value which is needed in that case is beforehand calculated by the decision value calculating means 1343. Here, the coordinate data after coordinate conversion is formed into a direct integer, and it memorizes to the integer coordinate table 37, and from this integral value, line width is computed and it memorizes to the horizon integer interval coordinate table 38. And the value (idyn) memorized on the horizon integer interval table 33 is compared with the value (idn) memorized to the horizon integer interval coordinate table 38, When having satisfied the conditions 40 that either of the white regions (idm-1, idm+1) of the both ends of a value (idm) which the value which fills idn>idyn in two or more black areas where line width is originally equal exists, and satisfies this condition 39 is 2 or more dots, the thicker compensation process 41 is performed. This example will perform thicker processing, if the line which compares a coordinate value when drawing without amending with the coordinate value after performing amendment by the conventional method, and can be amended more thickly exists. As for the interval between the contour coordinates which were the intervals same here before being integer-ized, it is needless to say that after integer-izing serves as the same interval.

[0027]Subsequently, the example of further others is described using drawing 3.

[0028]This example is the conditions of the beginning in said step ST26 (Sum of each integer line width of black area) + (white region x2) <= It is considered as whole length.

[0029]This is the conditions that the remainder which lengthened the sum of the integer line width of the black area in all the horizon integer interval tables 33 can secure the white region for 2 or more dots from whole length iys35. Since crushing of a character will not be caused even if it makes all the black areas increase by 1 dot temporarily if this condition can be fulfilled, thicker processing is attained. It has stopped only at judging whether the above-mentioned conditions can be processed thicker. In order to actually extract the line which needs thicker amendment, it is necessary to add conditions like the above-mentioned example further.

[0030]Next, the example in thicker processing is shown. This increases line width by 1 dot in said decision value calculating means to 1 set of lines of the line width same originally with which it was satisfied of the conditions that a line can be amended more thickly. About processings other than the above, it will carry out, for example in accordance with the conventional method like a JP,4-188190,A gazette. As for a coordinate point compensation process, it is needless to say that they may be methods other than the above.

[0031]According to these examples, a complicated character is amended thinness, in order to lose crushing, and the simple character can choose the correcting method for

which it was suitable for every character, such as amending more thickly, in order to prevent becoming weak-looking. As for the selection criterion of a correcting method, as these examples showed, it is needless to say that it may not be the above-mentioned method.

[0032]In the above, although only the processing to the y-coordinate of a horizon contour coordinate was explained, same processing is performed also about the x-coordinate of a perpendicular line contour coordinate.

[0033]

[Effect of the Invention]A correcting method is distinguished with the characteristic of a character, and the characteristic of a printer about the outline font which does not have the additional information for line width amendment in this invention so that clearly by old explanation, and line width amendment suitable for arbitrary characters and sizes is attained.

Therefore, it can amend by the optimal method corresponding to each character.

---

[Translation done.]